REMARKS

The present amendment responds to the Office Action mailed 28 November, 2006.

IN THE CLAIMS

Claims 19-30 were pending on the date of the Office Action. The present amendment cancels the pending claims 19-30 and replaces them with the new claims 31-45.

New independent claim 36

The new independent claim 36 is essentially based on former claims 7, 8, 9, and 10, and is directed to

> a method for enforcing a set of constraints that governs the integrity of information stored in a database system, the constraints being stored in a conceptual rules module *in the form of rules* for prescribing permitted states and transitions that the database can undertake (...)

It is directly derivable from the present specification that the constraints that are stored in the conceptual rules module are stored in the form of rules for prescribing permitted states and transitions that the database can undertake. Referring to the published version of the present patent application (US-2004/0107200), paragraph [0009], it is noted that a constraint should be considered as a special case of the term "conceptual rules", which are rules that prescribe all permitted states and transitions that the database can undertake. Further, from paragraph [0024], it is noted that the constraints are stored in a conceptual rules module, and that the conceptual rules module comprises rules for prescribing permitted states and transitions that the database can undertake.

According to Jacobs, "the non-uniqueness count is maintained in dynamic memory. The storage for the non-uniqueness count is allocated when the first duplicate entry is added to the uniqueness-required index" (Column 6, lines 40-44). Jacobs thus teaches that a non-uniqueness *count*, i.e. an incremented count of violations of the uniqueness constraint, is stored in a memory. Clearly, Jacobs does not disclose that constraints are stored in a conceptual rules module *in the form of rules* for prescribing permitted states and transitions that the database can undertake.

The method in claim 36 further comprises, i.a., the following step:

delaying constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check stack at the end of the transaction.

Jacobs does not disclose the step of executing *entries on the check stack* at the end of the transaction. In Jacobs, the memory or list contains counts of uniqueness constraint violations. The present check stack actually includes the checks, i.e. the rules, that are executed at the end of the transaction. Cf. also paragraph [0071] of the present application, which states that "the new stack is a list of all checks that has to be performed at the end of the transaction". Cf. also table B (paragraph [0072]) of the present application, which schematically illustrates an example of the stack content.

The method in claim 36 further comprises, i.a., the following step:

by a stack maker module operatively connected to a runtime module in said database system: receiving data from said runtime module, and creating and updating said check stack, and retrieving constraints from said conceptual rules module.

Since Jacobs does not disclose a conceptual rules module wherein constraints in the form of rules are stored, Jacobs evidently does not disclose the retrieving of constraints from a conceptual rules module either.

Dependent claims 37, 38

The dependent claims 37 and 38 specify additional features corresponding to the original claims 11 and 12, respectively.

Dependent claims 39, 40

The dependent claim 39 specifies constraints as transaction constraints, thus further distinguishing the claim from the disclosure of Jacobs, which merely mentions the enforcement of uniqueness constraints.

The dependent claim 40 specifies the constraints to be selected from: primary keys, foreign keys, subset constraints, and exclude constraints. cf. paragraph [0045] in the present patent application, thus further distinguishing the claim from the disclosure of Jacobs, which merely mentions the enforcement of uniqueness constraints.

New claims 31-35 and 41-45

New claims 31-35 correspond to the new claims 36-40, however, they are directed to a transaction based constraint enforcer rather than a method.

New claims 41-45 correspond to the new claims 31-35, however, they are directed to a database system rather than a constraint enforcer.

Analogous reasoning could therefore be carried out with respect to these groups of claims.

Conclusion

In view of the present amendment and the above remarks, allowance of the application is solicited.